UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/710,652	07/27/2004	Michael P. Quaranta	FIS920030351US1	4651

29505 7590 07/13/2007
LAW OFFICE OF DELIO & PETERSON, LLC.
121 WHITNEY AVENUE
NEW HAVEN, CT 06510

EXAMINER	
DWIVEDI, MAHESH H	
ART UNIT	PAPER NUMBER
2168	
MAIL DATE	DELIVERY MODE
07/13/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/710,652	**Applicant(s)** QUARANTA, MICHAEL P.	
	Examiner Mahesh H. Dwivedi	**Art Unit** 2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 April 2007*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-14 and 19-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-14 and 19-21* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *27 July 2004* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

Remarks

1. Applicant's election with traverse of Group I (Claims 1-3, 4-14, and 19-21) in the reply filed on 04/24/2007 is acknowledged. Group II, (Claims 17-18 and 22-23) is withdrawn from further consideration by the examiner, 37 CFR 1.142(b) as being drawn to a non-elected.

Specification

2. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the following is required: Dependent claim 16 recites "a command html for a collapsed section" and a "command html for an expanded section". The specification, however, fails to define what exactly the command html is referring to. For the purposes of examination of the instant application, the examiner considers a command html as a twistie.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claim 2 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The examiner notes that it is unclear as to what N in the phrase "1+N fetches" is referring to, since N is not defined in dependent claim 2, nor independent claim 1.

Claim 3 is rejected for incorporating the deficiencies of dependent claim 2.

5. Claim 6 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The examiner notes that it is unclear as to what N in the phrase "depth N expansion" is referring to, since N is not defined in dependent claim 6, nor independent claim 4. The examiner suggests that applicant specifically define "N" in the claims to clearly convey what the meaning of the variable is.

6. Claim 9 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The examiner is unclear as to what the cited html query is, since no explanation is given nor definition is given of it.

7. Claim 10 recites the limitation "by said comparison of said pre- and post-expanded section documents" in page 3. There is insufficient antecedent basis for this limitation in the claim, as independent claim 4 does not recite nor mention any comparison.

Claims 15-16 are rejected for incorporating the deficiencies of dependent claim 10.

8. Claim 10 recites the limitation "by said comparison of said pre- and post-expanded section documents" in page 3. There is insufficient antecedent basis for this limitation in the claim, as independent claim 4 does not recite nor mention any pre- and post- documents.

Claims 15-16 are rejected for incorporating the deficiencies of dependent claim 10.

9. Claim 12 recites the limitation "wherein said javascript conditionals include" in page 3. There is insufficient antecedent basis for this limitation in the claim, as independent claim 4 does not recite nor mention any javascript conditionals.

Claims 13-14 are rejected for incorporating the deficiencies of dependent claim 12.

10. Claim 15 recites the limitation "wherein said javascript comprises" in page 4. There is insufficient antecedent basis for this limitation in the claim, as independent claim 4 does not recite nor mention any javascript, nor does dependent claim 10.

Claim 16 is rejected for incorporating the deficiencies of dependent claim 15.

11. Claim 21 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The examiner notes that it is unclear as to what N in the phrase "depth N expansion" is referring to, since N is not defined in dependent claim 21,

nor independent claim 19. The examiner suggests that applicant specifically define “N” in the claims to clearly convey what the meaning of the variable is.

Claim Objections

12. Claim 7 is objected to because of the following informalities: The applicant is reminded that claims should not have quotation marks. The examiner suggests that applicant change “an **“expandection”** html query” to “an **expandsection** html query”. Appropriate correction is required.

Claims 8-9 are objected to for incorporating the deficiencies of dependent claim 7.

Claim 9 is objected to because of the following informalities: The applicant is reminded that claims should not have quotation marks. The examiner suggests that applicant change “**"/asdasd&ExpandSection=1,2,3,1.1"**” to “/asdasd&ExpandSection=1,2,3,1.1”. Appropriate correction is required.

Claim Rejections - 35 USC § 103

13. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

14. Claims 1-4, 12, and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Liman et al.** (U.S. Patent 6,078,327) in view of **Casahl** (Article entitled “Replic-Action Release 6.0”, dated April 1999).

15. Regarding claim 1, **Liman** teaches a method comprising:

B) creating a single document having all section data (Column 3, lines 7-18, Column 4, lines 1-9, lines 15-24, Figures 2a-2d); and

C) expanding and collapsing said section data at said client end (Column 4, lines 15-29, Figures 2a-2d).

The examiner notes that **Liman** teaches **“creating a single document having all section data”** as “The present invention is a Navigator utility or computer program that comprises a dockable window (miniframe window) that presents a collapse-expand tree control for traversing objects or elements within an application. In the preferred embodiment, a hierarchy of applications and objects are displayed in the dockable window, wherein various levels of the hierarchy may be displayed or hidden as desired and each level of the hierarchy is a different element of that level. The Navigator window allows a user to easily view the contents of all open or selected applications or objects and to navigate quickly to any application or object presented on the tree control" (Column 3, lines 7-18) and "The Navigator window 204 displays one or more applications, as well as one or more objects within the application, in a specified order. In the example of FIG. 2B using the APPROACH.RTM. application program, the application is identified and labeled as "c:.backslash.lotus.backslash.appl1.apr", wherein the ".apr" is an extension that identifies an APPROACH.RTM. application. Contained within this application are a number of object types are grouped into sections labeled as Tables, Forms, Worksheets, Crosstabs, Mailing Labels, Form Letters, Reports, Charts, Envelopes, Named Finds & Sorts, and Macros" (Column 4, lines 1-9). The examiner further notes that **Liman** teaches **"expanding and collapsing said section data at said client end"** as "A SingleClick on the triangle icon next to a label expands to the next level of the tree control or collapses the selected section. Thus, the triangle icon next to the label expands to the next level of the tree control or collapses the selected level. In the example of FIG. 2B, the triangle icon next to the application label has been selected to expand (as indicated by the downward-pointing triangle icon) to the next level of the tree control comprising the object types, which are collapsed in this view (as indicated by the rightward-pointing triangle icon)" (Column 4, lines 15-24).

Liman does not explicitly teach:

A) implementing a series of fetches for a Lotus Notes Domino document having a plurality section levels.

Casahl, however, teaches **"implementing a series of fetches for a Lotus Notes Domino document having a plurality section levels"** as "As a service for replication, Replic-Action can be used to access, transform, migrate, and synchronize data between virtually any environments, both workgroup and enterprise. To enumerate just a few of the possibilities, Replic-Action can create replications between:...Exchange and Notes" (Page 13), "Replic-Action provides the tools to support migration from one environment to another" (Page 21), and ""Applications connecting to groupware environments like Lotus Notes and Microsoft Exchange can make use of the new collapse/expand feature to collapse multiple line item detail records in relational databases to multi-valued lists in groupware, and vice versa" (Pages 27-28).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Casahl's** would have allowed **Liman's** to provide a method for users in Lotus environments and Microsoft exchange environments to continue work while exchanging information with each other, either in real0time or via replication, as noted by **Casahl** (Page 21).

Regarding claim 2, **Liman** does not explicitly teach a method comprising:

A) including performing 1+N fetches, one for each level of sections.

Casahl, however, teaches **"including performing 1+N fetches, one for each level of sections"** as "As a service for replication, Replic-Action can be used to access, transform, migrate, and synchronize data between virtually any environments, both workgroup and enterprise. To enumerate just a few of the possibilities, Replic-Action can create replications between:...Exchange and Notes" (Page 13), "Replic-Action provides the tools to support migration from one environment to another" (Page 21), and ""Applications connecting to groupware environments like Lotus Notes and Microsoft Exchange can make use of the new collapse/expand feature to collapse multiple line item detail records in relational databases to multi-valued lists in groupware, and vice versa" (Pages 27-28).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Casahl's** would have allowed **Liman's** to provide a method for users in Lotus environments and Microsoft exchange environments to continue work while exchanging information with each other, either in real0time or via replication, as noted by **Casahl** (Page 21).

Regarding claim 3, **Liman** further teaches a method comprising:

A) wherein said single document includes a plurality of pages equal to said plurality of section levels (Column 3, lines 7-18, Column 4, lines 1-9, lines 15-24, Figures 2a-2d).

The examiner notes that **Liman** teaches **"wherein said single document includes a plurality of pages equal to said plurality of section levels"** as "The present invention is a Navigator utility or computer program that comprises a dockable window (miniframe window) that presents a collapse-expand tree control for traversing objects or elements within an application. In the preferred embodiment, a hierarchy of applications and objects are displayed in the dockable window, wherein various levels of the hierarchy may be displayed or hidden as desired and each level of the hierarchy is a different element of that level. The Navigator window allows a user to easily view the contents of all open or selected applications or objects and to navigate quickly to any application or object presented on the tree control" (Column 3, lines 7-18) and "The Navigator window 204 displays one or more applications, as well as one or more objects within the application, in a specified order. In the example of FIG. 2B using the APPROACH.RTM. application program, the application is identified and labeled as "c:.backslash.lotus.backslash.appl1.apr", wherein the ".apr" is an extension that identifies an APPROACH.RTM. application. Contained within this application are a number of object types are grouped into sections labeled as Tables, Forms, Worksheets, Crosstabs, Mailing Labels, Form Letters, Reports, Charts, Envelopes, Named Finds & Sorts, and Macros" (Column 4, lines 1-9)

Regarding claim 4, **Liman** teaches a method comprising:

B) performing a recursive differential process step (Column 3, lines 7-18, Column 4, lines 1-9, lines 15-24, Figures 2a-2d); and

C) expanding and collapsing said section data at said client end (Column 4, lines 15-29, Figures 2a-2d).

The examiner notes that **Liman** teaches **"performing a recursive differential process step"** as "The present invention is a Navigator utility or computer program that comprises a dockable window (miniframe window) that presents a collapse-expand tree control for traversing objects or elements within an application. In the preferred embodiment, a hierarchy of applications and objects are displayed in the dockable window, wherein various levels of the hierarchy may be displayed or hidden as desired and each level of the hierarchy is a different element of that level. The Navigator window allows a user to easily view the contents of all open or selected applications or objects and to navigate quickly to any application or object presented on the tree control" (Column 3, lines 7-18) and "The Navigator window 204 displays one or more applications, as well as one or more objects within the application, in a specified order. In the example of FIG. 2B using the APPROACH.RTM. application program, the application is identified and labeled as "c:.backslash.lotus.backslash.appl1.apr", wherein the ".apr" is an extension that identifies an APPROACH.RTM. application. Contained within this application are a number of object types are grouped into sections labeled as Tables, Forms, Worksheets, Crosstabs, Mailing Labels, Form Letters, Reports, Charts, Envelopes, Named Finds & Sorts, and Macros" (Column 4, lines 1-9). The examiner further notes that **Liman** teaches **"expanding and collapsing said section data at said client end"** as "A SingleClick on the triangle icon next to a label expands to the next level of the tree control or collapses the selected section. Thus, the triangle icon next to the label expands to the next level of the tree control or collapses the selected level. In the example of FIG. 2B, the triangle icon next to the application label has been selected to expand (as indicated by the downward-pointing triangle icon) to the next level of the tree control comprising the object types, which are collapsed in this view (as indicated by the rightward-pointing triangle icon)" (Column 4, lines 15-24).

Liman does not explicitly teach:

A) performing iterative recursive intelligent fetch process steps.

Casahl, however, teaches **"performing iterative recursive intelligent fetch process steps"** as "As a service for replication, Replic-Action can be used to access, transform, migrate, and synchronize data between virtually any environments, both workgroup and enterprise. To enumerate just a few of the possibilities, Replic-Action can create replications between:...Exchange and Notes" (Page 13), "Replic-Action provides the tools to support migration from one environment to another" (Page 21), and "Applications connecting to groupware environments like Lotus Notes and Microsoft Exchange can make use of the new collapse/expand feature to collapse multiple line item detail records in relational databases to multi-valued lists in groupware, and vice versa" (Pages 27-28).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Casahl's** would have allowed **Liman's** to provide a method for users in Lotus environments and Microsoft exchange environments to continue work while exchanging information with each other, either in real0time or via replication, as noted by **Casahl** (Page 21).

Regarding claim 12, **Liman** teaches a method comprising:

A) wherein said javascript conditionals include instructions to display section data (Column 3, lines 7-18, Column 4, lines 1-9, lines 15-24, Figures 2a-2d).

The examiner notes that **Liman** teaches **"wherein said javascript conditionals include instructions to display section data"** as "The present invention is a Navigator utility or computer program that comprises a dockable window (miniframe window) that presents a collapse-expand tree control for traversing objects or elements within an application. In the preferred embodiment, a hierarchy of applications and objects are displayed in the dockable window, wherein various levels of the hierarchy may be displayed or hidden as desired and each level of the hierarchy is a different element of that level. The Navigator window allows a user to easily view the contents of all open or selected applications or objects and to navigate quickly to any application or object presented on the tree control" (Column 3, lines 7-18) and "The Navigator window 204 displays one or more applications, as well as one or more objects within the application, in a specified order. In the example of FIG. 2B using the APPROACH.RTM. application program, the application is identified and labeled as "c:.backslash.lotus.backslash.appl1.apr", wherein the ".apr" is an extension that identifies an APPROACH.RTM. application. Contained within this application are a number of object types are grouped into sections labeled as Tables, Forms, Worksheets, Crosstabs, Mailing Labels, Form Letters, Reports, Charts, Envelopes, Named Finds & Sorts, and Macros" (Column 4, lines 1-9).

Regarding claim 19, **Liman** teaches a program storage device comprising:
B) performing a recursive differential process step (Column 3, lines 7-18, Column 4, lines 1-9, lines 15-24, Figures 2a-2d); and
C) expanding and collapsing said section data at said client end (Column 4, lines 15-29, Figures 2a-2d).

The examiner notes that **Liman** teaches **"performing a recursive differential process step"** as "The present invention is a Navigator utility or computer program that comprises a dockable window (miniframe window) that presents a collapse-expand tree control for traversing objects or elements within an application. In the preferred embodiment, a hierarchy of applications and objects are displayed in the dockable window, wherein various levels of the hierarchy may be displayed or hidden as desired and each level of the hierarchy is a different element of that level. The Navigator window allows a user to easily view the contents of all open or selected applications or objects and to navigate quickly to any application or object presented on the tree control" (Column 3, lines 7-18) and "The Navigator window 204 displays one or more applications, as well as one or more objects within the application, in a specified order. In the example of FIG. 2B using the APPROACH.RTM. application program, the application is identified and labeled as "c:.backslash.lotus.backslash.appl1.apr", wherein the ".apr" is an extension that identifies an APPROACH.RTM. application. Contained within this application are a number of object types are grouped into sections labeled as Tables, Forms, Worksheets, Crosstabs, Mailing Labels, Form Letters, Reports, Charts, Envelopes, Named Finds & Sorts, and Macros" (Column 4, lines 1-9). The examiner further notes that **Liman** teaches **"expanding and collapsing said section data at said client end"** as "A SingleClick on the triangle icon next to a label expands to the next level of the tree control or collapses the selected section. Thus, the triangle icon next to the label expands to the next level of the tree control or collapses the selected level. In the example of FIG. 2B, the triangle icon next to the application label has been selected to expand (as indicated by the downward-pointing triangle icon) to the next level of the tree control comprising the object types, which are collapsed in this view (as indicated by the rightward-pointing triangle icon)" (Column 4, lines 15-24).

Liman does not explicitly teach:

A) performing iterative recursive intelligent fetch process steps.

Casahl, however, teaches **"performing iterative recursive intelligent fetch process steps"** as "As a service for replication, Replic-Action can be used to access, transform, migrate, and synchronize data between virtually any environments, both workgroup and enterprise. To enumerate just a few of the possibilities, Replic-Action can create replications between:...Exchange and Notes" (Page 13), "Replic-Action provides the tools to support migration from one environment to another" (Page 21), and "Applications connecting to groupware environments like Lotus Notes and Microsoft Exchange can make use of the new collapse/expand feature to collapse multiple line item detail records in relational databases to multi-valued lists in groupware, and vice versa" (Pages 27-28).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Casahl's** would have allowed **Liman's** to provide a method for users in Lotus environments and Microsoft exchange environments to continue work while exchanging information with each other, either in real0time or via replication, as noted by **Casahl** (Page 21).

16. Claims 5, 7-9, 11, and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Liman et al.** (U.S. Patent 6,078,327) in view of **Casahl** (Article entitled "Replic-Action Release 6.0", dated April 1999) as applied to claims -4, 12, and 19, and further in view of **Kaczmarkski et al.** (U.S. Patent 6,314,424).

17. Regarding claim 5, **Liman** does not explicitly teach a method comprising:

A) wherein said iterative recursive intelligent fetch process includes: parsing said Domino document.

Casahl, however, teaches **"wherein said iterative recursive intelligent fetch process includes: parsing said Domino document"** as "As a service for replication, Replic-Action can be used to access, transform, migrate, and synchronize data between virtually any environments, both workgroup and enterprise. To enumerate just a few of the possibilities, Replic-Action can create replications between:...Exchange and Notes" (Page 13), "Replic-Action provides the tools to support migration from one environment to another" (Page 21), and "Applications connecting to groupware environments like Lotus Notes and Microsoft Exchange can make use of the new collapse/expand feature to collapse multiple line item detail records in relational databases to multi-valued lists in groupware, and vice versa" (Pages 27-28).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Casahl's** would have allowed **Liman's** to provide a method for users in Lotus environments and Microsoft exchange environments to continue work while exchanging information with each other, either in real0time or via replication, as noted by **Casahl** (Page 21).

Liman and **Casahl** do not explicitly teach:

B) identifying unexpanded sections; and

C) constructing a URL to expand said unexpanded sections.

Kaczmarski, however, teaches **"identifying unexpanded sections"** as "The linked list is then passed back to the tree HTML page generation engine. The tree HTML page generation engine turns the linked list into an HTML page containing nodes and leaves. Each node has an embedded URL associated with it and contains a special encoding to memorize the tree expansion state information. The generated tree view is then returned to the web server and then to the browser" (Column 2, lines 35-42) and "The tree is shown in its fully-extended format. The bit string representation for this would be the following: 1111. The first 1 represents the expanded state for the A node. Since there are no expandable nodes below A no more bits are needed to represent the

A subtree. The second 1 represents the expansion state of tree B. Note that tree B has 2 subtrees that can be in an expanded or collapsed state. The two remaining 1's indicate the expanded state of the B's two subtrees B1 and B2. If there were a top node C we would still know that the 3rd and 4th 1's correspond to B's subtrees because we know that B has 2 subtrees (this is fixed in the application--deterministic). If the given tree was contracted at node A and node B1, the visual affect would be the following:...The bit string representation would be: 0101" (Column 4, lines 66-67-Column 5, lines 1-20), and **"constructing a URL to expand said unexpanded sections"** as "The linked list is then passed back to the tree HTML page generation engine. The tree HTML page generation engine turns the linked list into an HTML page containing nodes and leaves. Each node has an embedded URL associated with it and contains a special encoding to memorize the tree expansion state information. The generated tree view is then returned to the web server and then to the browser" (Column 2, lines 35-42) and "The tree is shown in its fully-extended format. The bit string representation for this would be the following: 1111. The first 1 represents the expanded state for the A node. Since there are no expandable nodes below A no more bits are needed to represent the A subtree. The second 1 represents the expansion state of tree B. Note that tree B has 2 subtrees that can be in an expanded or collapsed state. The two remaining 1's indicate the expanded state of the B's two subtrees B1 and B2. If there were a top node C we would still know that the 3rd and 4th 1's correspond to B's subtrees because we know that B has 2 subtrees (this is fixed in the application--deterministic). If the given tree was contracted at node A and node B1, the visual affect would be the following:...The bit string representation would be: 0101" (Column 4, lines 66-67-Column 5, lines 1-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Kaczmarski's** would have allowed **Liman's** and **Casahl's** to provide a method for users to access internet information in an efficient and rapid access while maintaining real-time needs, as noted by **Kaczmarski** (Column 1, lines 57-61).

Regarding claim 7, **Liman** and **Casahl** do not explicitly teach a method comprising:

A) instructing Domino to expand each section by an "expandsection" query.

Kaczmarski, however, teaches **"instructing Domino to expand each section by an "expandsection" query"** as "The linked list is then passed back to the tree HTML page generation engine. The tree HTML page generation engine turns the linked list into an HTML page containing nodes and leaves. Each node has an embedded URL associated with it and contains a special encoding to memorize the tree expansion state information. The generated tree view is then returned to the web server and then to the browser" (Column 2, lines 35-42) and "The expansion and contraction states for the navagation tree is maintained as a list of 0's and 1's imbedded as a certain component in the HTML URL. 1's represent "open" tree branches (where subtree elements are shown) and 0's represent closed branches (where subtree elements are not shown. Because the number of sub-elements for any given internal branch are known, the number of 0's and 1's that follow an open branch are deterministic. This simple encoding allows one to understand the current expansion state of the tree given the list of 0's and 1's An example follows...The tree is shown in its fully-extended format" (Column 4, lines 43-66).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Kaczmarski's** would have allowed **Liman's** and **Casahl's** to provide a method for users to access internet information in an efficient and rapid access while maintaining real-time needs, as noted by **Kaczmarski** (Column 1, lines 57-61).

Regarding claim 8, **Liman** and **Casahl** do not explicitly teach a method comprising:

A) wherein said expandsection html query includes said html query for a plurality of section expansions.

Kaczmarski, however, teaches **"wherein said expandsection html query includes said html query for a plurality of section expansions"** as "The linked list is then passed back to the tree HTML page generation engine. The tree HTML page generation engine turns the linked list into an HTML page containing nodes and leaves. Each node has an embedded URL associated with it and contains a special encoding to memorize the tree expansion state information. The generated tree view is then returned to the web server and then to the browser" (Column 2, lines 35-42) and "The expansion and contraction states for the navagation tree is maintained as a list of 0's and 1's imbedded as a certain component in the HTML URL. 1's represent "open" tree branches (where subtree elements are shown) and 0's represent closed branches (where subtree elements are not shown. Because the number of sub-elements for any given internal branch are known, the number of 0's and 1's that follow an open branch are deterministic. This simple encoding allows one to understand the current expansion state of the tree given the list of 0's and 1's An example follows...The tree is shown in its fully-extended format" (Column 4, lines 43-66).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Kaczmarski's** would have allowed **Liman's** and **Casahl's** to provide a method for users to access internet information in an efficient and rapid access while maintaining real-time needs, as noted by **Kaczmarski** (Column 1, lines 57-61).

Regarding claim 9, **Liman** and **Casahl** do not explicitly teach a method comprising:

A) including said html query of a form: "/asdasd&ExpandSection=1,2,3,1.1".

Kaczmarski, however, teaches **"including said html query of a form: "/asdasd&ExpandSection=1,2,3,1.1""** as "The linked list is then passed back to the tree HTML page generation engine. The tree HTML page generation engine turns the linked list into an HTML page containing nodes and leaves. Each node has an embedded URL associated with it and contains a special encoding to memorize the tree expansion state information. The generated tree view is then returned to the web server and then to the browser" (Column 2, lines 35-42) and "The expansion and contraction states for the navagation tree is maintained as a list of 0's and 1's imbedded as a certain component in the HTML URL. 1's represent "open" tree branches (where subtree elements are shown) and 0's represent closed branches (where subtree elements are not shown. Because the number of sub-elements for any given internal branch are known, the number of 0's and 1's that follow an open branch are deterministic. This simple encoding allows one to understand the current expansion state of the tree given the list of 0's and 1's An example follows...The tree is shown in its fully-extended format" (Column 4, lines 43-66).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Kaczmarski's** would have allowed **Liman's** and **Casahl's** to provide a method for users to access internet information in an efficient and rapid access while maintaining real-time needs, as noted by **Kaczmarski** (Column 1, lines 57-61).

Regarding claim 11, **Liman** and **Casahl** do not explicitly teach a method comprising:

A) wherein said parsing said Domino document includes: performing an iterative process to identify all subsections of said document, comprising: fetching a document with all sections collapsed and identifying sections having expansion;

B) if level 1 section expansions are identified, fetching said document having level 1 sections expanded;

C) discovering and parsing new sub-sections below said level 1 sections;

D) if level 2 section expansions are identified, fetching said document having level 2 sections expanded;

E) discovering and parsing new sub-sections below said level 2 sections; and

F) continuing said iterative process until level N section expansions are identified and fetched.

Kaczmarski, however, teaches **"wherein said parsing said Domino document includes: performing an iterative process to identify all subsections of said document, comprising: fetching a document with all sections collapsed and identifying sections having expansion"** as "The linked list is then passed back to the tree HTML page generation engine. The tree HTML page generation engine turns the linked list into an HTML page containing nodes and leaves. Each node has an embedded URL associated with it and contains a special encoding to memorize the tree expansion state information. The generated tree view is then returned to the web server and then to the browser" (Column 2, lines 35-42) and "The expansion and contraction states for the navagation tree is maintained as a list of 0's and 1's imbedded as a certain component in the HTML URL. 1's represent "open" tree branches (where subtree elements are shown) and 0's represent closed branches (where subtree elements are not shown. Because the number of sub-elements for any given internal branch are known, the number of 0's and 1's that follow an open branch are deterministic. This simple encoding allows one to understand the current expansion state of the tree given the list of 0's and 1's An example follows...The tree is shown in its fully-extended format" (Column 4, lines 43-66), **"if level 1 section expansions are identified, fetching said document having level 1 sections expanded"** as "The linked list is then passed back to the tree HTML page generation engine. The tree HTML page generation engine turns the linked list into an HTML page containing nodes and leaves. Each node has an embedded URL associated with it and contains a special encoding to memorize the tree expansion state information. The generated tree view is then returned to the web server and then to the browser" (Column 2, lines 35-42) and "The expansion and contraction states for the navagation tree is maintained as a list of 0's and 1's imbedded as a certain component in the HTML URL. 1's represent "open" tree branches (where subtree elements are shown) and 0's represent closed branches (where subtree elements are not shown. Because the number of sub-elements for any given internal branch are known, the number of 0's and 1's that follow an open branch are deterministic. This simple encoding allows one to understand the current expansion state of the tree given the list of 0's and 1's An example follows...The tree is shown in its fully-extended format" (Column 4, lines 43-66), **"discovering and parsing new sub-sections below said level 1 sections"** as "The linked list is then passed back to the tree HTML page generation engine. The tree HTML page generation engine turns the linked list into an HTML page containing nodes and leaves. Each node has an embedded URL associated with it and contains a special encoding to memorize the tree expansion state information. The generated tree view is then returned to the web server and then to the browser" (Column 2, lines 35-42) and "The expansion and contraction states for the navagation tree is maintained as a list of 0's and 1's imbedded as a certain component in the HTML URL. 1's represent "open" tree branches (where subtree elements are shown) and 0's represent closed branches (where subtree elements are not shown. Because the number of sub-elements for any given internal branch are known, the number of 0's and 1's that follow an open branch are deterministic. This simple encoding allows one to understand the current expansion state of the tree given the list of 0's and 1's An example follows...The tree is shown in its fully-extended format" (Column 4, lines 43-66), **"if level 2 section expansions are identified, fetching said document having level 2 sections expanded"** as "The linked list is then passed back to the tree HTML page generation engine. The tree HTML page generation engine turns the linked list into an HTML page containing nodes and leaves. Each node has an embedded URL associated with it and contains a special encoding to memorize the tree expansion state information. The generated tree view is then returned to the web server and then to the browser" (Column 2, lines 35-42) and "The expansion and contraction states for the navagation tree is maintained as a list of 0's and 1's imbedded as a certain component in the HTML URL. 1's represent "open" tree branches (where subtree elements are shown) and 0's represent closed branches (where subtree elements are not shown. Because the number of sub-elements for any given internal branch are known, the number of 0's and 1's that follow an open branch are deterministic. This simple encoding allows one to understand the current expansion state of the tree given the list of 0's and 1's An example follows...The tree is shown in its fully-extended format" (Column 4, lines 43-66), **"discovering and parsing new sub-sections below said level 2 sections"** as "The linked list is then passed back to the tree HTML page generation engine. The tree HTML page generation engine turns the linked list into an HTML page containing nodes and leaves. Each node has an embedded URL associated with it and contains a special encoding to memorize the tree expansion state information. The generated tree view is then returned to the web server and then to the browser" (Column 2, lines 35-42) and "The expansion and contraction states for the navagation tree is maintained as a list of 0's and 1's imbedded as a certain component in the HTML URL. 1's represent "open" tree branches (where subtree elements are shown) and 0's represent closed branches (where subtree elements are not shown. Because the number of sub-elements for any given internal branch are known, the number of 0's and 1's that follow an open branch are deterministic. This simple encoding allows one to understand the current expansion state of the tree given the list of 0's and 1's An example follows...The tree is shown in its fully-extended format" (Column 4, lines 43-66), and **"continuing said iterative process until level N section expansions are identified and fetched"** as "The linked list is then passed back to the tree HTML page generation engine. The tree HTML page generation engine turns the linked list into an HTML page containing nodes and leaves. Each node has an embedded URL associated with it and contains a special encoding to memorize the tree expansion state information. The generated tree view is then returned to the web server and then to the browser" (Column 2, lines 35-42) and "The expansion and contraction states for the navagation tree is maintained as a list of 0's and 1's imbedded as a certain component in the HTML URL. 1's represent "open" tree branches (where subtree elements are shown) and 0's represent closed branches (where subtree elements are not shown. Because the number of sub-elements for any given internal branch are known, the number of 0's and 1's that follow an open branch are deterministic. This simple encoding allows one to understand the current expansion state of the tree given the list of 0's and 1's An example follows...The tree is shown in its fully-extended format" (Column 4, lines 43-66).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Kaczmarski's** would have allowed **Liman's** and **Casahl's** to provide a method for users to access internet information in an efficient and rapid access while maintaining real-time needs, as noted by **Kaczmarski** (Column 1, lines 57-61).

Regarding claim 20, **Liman** does not explicitly teach a program storage device comprising:

A) wherein said iterative recursive intelligent fetch process includes: parsing said Domino document.

Casahl, however, teaches **"wherein said iterative recursive intelligent fetch process includes: parsing said Domino document"** as "As a service for replication, Replic-Action can be used to access, transform, migrate, and synchronize data between virtually any environments, both workgroup and enterprise. To enumerate just a few of the possibilities, Replic-Action can create replications between:...Exchange and Notes" (Page 13), "Replic-Action provides the tools to support migration from one environment to another" (Page 21), and "Applications connecting to groupware environments like Lotus Notes and Microsoft Exchange can make use of the new collapse/expand feature to collapse multiple line item detail records in relational databases to multi-valued lists in groupware, and vice versa" (Pages 27-28).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Casahl's** would have allowed **Liman's** to provide a method for users in Lotus environments and Microsoft exchange environments to continue work while exchanging information with each other, either in real0time or via replication, as noted by **Casahl** (Page 21).

Liman and **Casahl** do not explicitly teach:

B) identifying unexpanded sections; and

C) constructing a URL to expand said unexpanded sections.

Kaczmarski, however, teaches **"identifying unexpanded sections"** as "The linked list is then passed back to the tree HTML page generation engine. The tree HTML page generation engine turns the linked list into an HTML page containing nodes and leaves. Each node has an embedded URL associated with it and contains a special encoding to memorize the tree expansion state information. The generated tree view is then returned to the web server and then to the browser" (Column 2, lines 35-42) and "The tree is shown in its fully-extended format. The bit string representation for this would be the following: 1111. The first 1 represents the expanded state for the A node. Since there are no expandable nodes below A no more bits are needed to represent the A subtree. The second 1 represents the expansion state of tree B. Note that tree B has 2 subtrees that can be in an expanded or collapsed state. The two remaining 1's indicate the expanded state of the B's two subtrees B1 and B2. If there were a top node C we would still know that the 3rd and 4th 1's correspond to B's subtrees because we know that B has 2 subtrees (this is fixed in the application--deterministic). If the given tree was contracted at node A and node B1, the visual affect would be the following:...The bit string representation would be: 0101" (Column 4, lines 66-67-Column 5, lines 1-20), and **"constructing a URL to expand said unexpanded sections"** as "The linked list is then passed back to the tree HTML page generation engine. The tree HTML page generation engine turns the linked list into an HTML page containing nodes and leaves. Each node has an embedded URL associated with it and contains a special encoding to memorize the tree expansion state information. The generated tree view is then returned to the web server and then to the browser" (Column 2, lines 35-42) and "The tree is shown in its fully-extended format. The bit string representation for this would be the following: 1111. The first 1 represents the expanded state for the A node. Since there are no expandable nodes below A no more bits are needed to represent the A subtree. The second 1 represents the expansion state of tree B. Note that tree B has 2 subtrees that can be in an expanded or collapsed state. The two remaining 1's indicate the expanded state of the B's two subtrees B1 and B2. If there were a top node C we would still know that the 3rd and 4th 1's correspond to B's subtrees because we know that B has 2 subtrees (this is fixed in the application--deterministic). If the given tree was contracted at node A and node B1, the visual affect would be the following:...The bit string representation would be: 0101" (Column 4, lines 66-67-Column 5, lines 1-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Kaczmarski's** would have allowed **Liman's** and **Casahl's** to provide a method for users to access internet information in an efficient and rapid access while maintaining real-time needs, as noted by **Kaczmarski** (Column 1, lines 57-61).

18. Claims 10, and 13-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Liman et al.** (U.S. Patent 6,078,327) in view of **Casahl** (Article entitled "Replic-Action Release 6.0", dated April 1999) as applied to claims 1-4, 12, and 19, and further in view of **Baudisch et al.** (Article entitled "Callapse-to-Zoom: Viewing Web pages on Small Screen Devices by Interactively Removing Irrelevant Content, dated 07/17/2004").

19. Regarding claim 10, **Liman** and **Casahl** do not explicitly teach a method comprising:

A) including identifying data for each of said sections by said comparison of said pre- and post-expanded section documents.

Baudisch, however, teaches **"including identifying data for each of said sections by said comparison of said pre- and post-expanded section documents"** as "The next day, the user invokes the bookmark created earlier and the browser loads the new edition of the page. However, the browser also restores the collapse-state the page was in when the bookmark was created. All headlines are therefore readable right away; no further user interaction is required" (Page 3, section h).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Baudisch's** would have allowed **Liman's** and **Casahl's** to provide a method for users to view and load content faster by setting content to user preferences, as noted by **Baudisch** (Page 3, section h).

Regarding claim 13, **Liman** and **Casahl** do not explicitly teach a method comprising:

A) including using cookies to preserve each expansion state.

Baudisch, however, teaches **"including using cookies to preserve each expansion state"** as "The next day, the user invokes the bookmark created earlier and the browser loads the new edition of the page. However, the browser also restores the collapse-state the page was in when the bookmark was created. All headlines are therefore readable right away; no further user interaction is required" (Page 3, section h).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Baudisch's** would have allowed **Liman's** and **Casahl's** to provide a method for users to view and load content faster by setting content to user preferences, as noted by **Baudisch** (Page 3, section h).

Regarding claim 14, **Liman** further teaches a method comprising:

A) including modifying said expansion state by having a user click on a triangular twisty (Column 4, lines 15-29, Figures 2a-2d).

The examiner notes that **Liman** teaches **"including modifying said expansion state by having a user click on a triangular twisty"** as "A SingleClick on the triangle icon next to a label expands to the next level of the tree control or collapses the selected section. Thus, the triangle icon next to the label expands to the next level of the tree control or collapses the selected level. In the example of FIG. 2B, the triangle icon next to the application label has been selected to expand (as indicated by the downward-pointing triangle icon) to the next level of the tree control comprising the object types, which are collapsed in this view (as indicated by the rightward-pointing triangle icon)" (Column 4, lines 15-24).

Regarding claim 15, **Liman** and **Casahl** do not explicitly teach a method comprising:

A) wherein said javascript comprises: a session cookie remembering when said section was expanded;

B) a set cookie to remember when said section is collapsed; and

C) a set cookie to remember when said section is currently expanded.

Baudisch, however, teaches **"wherein said javascript comprises: a session cookie remembering when said section was expanded"** as "The next day, the user invokes the bookmark created earlier and the browser loads the new edition of the page. However, the browser also restores the collapse-state the page was in when the bookmark was created. All headlines are therefore readable right away; no further user interaction is required" (Page 3, section h), **"a set cookie to remember when said section is collapsed"** as "The next day, the user invokes the bookmark created earlier and the browser loads the new edition of the page. However, the browser also restores the collapse-state the page was in when the bookmark was created. All headlines are therefore readable right away; no further user interaction is required" (Page 3, section h), and **"a set cookie to remember when said section is currently expanded"** as "The next day, the user invokes the bookmark created earlier and the browser loads the new edition of the page. However, the browser also restores the collapse-state the page was in when the bookmark was created. All headlines are therefore readable right away; no further user interaction is required" (Page 3, section h).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Baudisch's** would have allowed **Liman's** and **Casahl's** to provide a method for users to view and load content faster by setting content to user preferences, as noted by **Baudisch** (Page 3, section h).

Regarding claim 16, **Liman** further teaches a method comprising:

A) a command html for a collapsed section and a command html for an expanded section (Column 4, lines 15-29, Figures 2a-2d).

The examiner notes that **Liman** teaches **"a command html for a collapsed section and a command html for an expanded section"** as "A SingleClick on the triangle icon next to a label expands to the next level of the tree control or collapses the selected section. Thus, the triangle icon next to the label expands to the next level of the tree control or collapses the selected level. In the example of FIG. 2B, the triangle icon next to the application label has been selected to expand (as indicated by the downward-pointing triangle icon) to the next level of the tree control comprising the object types, which are collapsed in this view (as indicated by the rightward-pointing triangle icon)" (Column 4, lines 15-24).

Allowable Subject Matter

20. Claims 6 and 21 would be allowable if rewritten to overcome the rejection(s) under 35 U.S.C. 112, 2nd paragraph, set forth in this Office action and to include all of the limitations of the base claim and any intervening claims

Conclusion

21. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Patent 6,574,644 issued to **Hsu et al.** on 03 June 2004. The subject matter disclosed therein is pertinent to that of claims 1-14, and 19-21 (e.g., methods to migrate lotus documents).

U.S. PGPUB 2003/0202019 issued to **Detweiler et al.** on 30 October 2003. The subject matter disclosed therein is pertinent to that of claims 1-14, and 19-21 (e.g., methods to migrate lotus documents).

U.S. PGPUB 2005/0198582 issued to **Hennum et al.** on 08 September 2005. The subject matter disclosed therein is pertinent to that of claims 1-14, and 19-21 (e.g., methods to migrate lotus documents).

U.S. PGPUB 2002/0099723 issued to **Chiesa** on 25 July 2002. The subject matter disclosed therein is pertinent to that of claims 1-14, and 19-21 (e.g., methods to migrate lotus documents).

Article entitled "Using Transend Migrator for Lotus Notes data to Microsoft Outlook/Exchange", by **Transend**, dated 28 June 2004. The subject matter disclosed therein is pertinent to that of claims 1-14, and 19-21 (e.g., methods to migrate lotus documents).

Article entitled "Collaborative Application Integration, Extension, and Migration with Casahl ecKnowledge", by **Casahl**, dated January 2004. The subject matter disclosed therein is pertinent to that of claims 1-14, and 19-21 (e.g., methods to migrate lotus documents).

Contact Information

22. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mahesh Dwivedi whose telephone number is (571) 272-2731. The examiner can normally be reached on Monday to Friday 8:20 am – 4:40 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached (571) 272-3642. The fax number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mahesh Dwivedi
Patent Examiner
Art Unit 2168

July 06, 2007

MOHAMMAD ALI
PRIMARY EXAMINER